Содержание

У вас не получается читать/выводить данные?
Воспользуйтесь примерами (c++) (python).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на `GNU C++` компиляторы с суффиксом `inc`.
Подни можно пользоваться дополнительной библиотекой (`optimization.h`).
То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.
И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): пример.

Для тех, кто хочет разобраться, как всё это работает.
Короткая версия быстрого ввода-вывода (тык) и короткая версия аллокатора (тык).

Must have

Задача 11A. Выбор вершин взвешенного дерева [0.1 sec, 256 mb]

Дан граф, являющийся деревом. В вершинах графа написаны целые числа. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины i, не являющейся корнем, задан номер вершины-предка p_i в корневом дереве. Дерево, заданное таким образом, состоит из рёбер $i - p_i$ для всех вершин i, кроме корня.

В первой строке входного файла записано целое число n — количество вершин в графе ($1 \leqslant n \leqslant 100$). В следующих n строках задан граф. В i-й из этих строк записаны через пробел два целых числа p_i и q_i; здесь p_i — номер вершины-предка i-ой вершины, а q_i — число, записанное в этой вершине. Для корня дерева $p_i = 0$; для всех остальных вершин $1 \leqslant p_i \leqslant n$. Числа q_i не превосходят по модулю 10 000.

Гарантируется, что заданный во входном файле граф является деревом.

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальную сумму чисел в допустимом множестве.

Примеры

stdin	stdout
5 0 1 1 2 1 3 2 4 3 5	10
6 5 8 6 0 5 -1 1 1 0 3 1 2	8

Замечание

Чтобы хранить дерево, используйте `vector<int> c[n];`

Задача 11B. Строка Фибоначчи [0.1 sec, 256 mb]

Строка Фибоначчи — это строка из нулей и единиц, в которой не встречается двух идущих подряд единиц.

Даны числа n и k. Нужно вывести строку Фибоначчи, которая состоит из n цифр и является k-ой в лексикографическом порядке из таких строк.

Формат входных данных

Целые числа n и k ($0 \leqslant n \leqslant 44$, $0 \leqslant k \leqslant 2 \cdot 10^9$).

Гарантируется, что k-ая строка из n символов существует. Строки нумеруются с нуля.

Формат выходных данных

Выведите лексикографически k-ую строку Фибоначчи длины n.

Вывод *обязательно* завершать переводом строки.

Примеры

stdin	stdout
3 0	000
3 1	001
3 2	010
3 3	100
3 4	101

Замечание

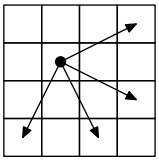
В тему лекции.

Обязательные задачи

Задача 11C. Ход конём - 2 [0.1 sec, 256 mb]

Дана прямоугольная доска $N \times M$ (N строк и M столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

Формат входных данных

Два натуральных числа N и M ($1 \leqslant N, M \leqslant 50$).

Формат выходных данных

Количество способов добраться конём до правого нижнего угла доски.

Пример

stdin	stdout
4 4	2
15 14	7884330

Замечание

`__int128` или написать ручками сложение... (один цикл `for`).

Задача 11E. Палиндром [0.2 sec, 256 mb]

Палиндромом называется строка, которая читается одинаково как слева направо, так и справа налево. Требуется найти самый длинный палиндром P, получающийся из данной строки S удалением любого (возможно, нулевого) количества символов.

Формат входных данных

Входной файл содержит строчку S, состоящую из строчных латинских букв (a–z).

Длина S не превышает 1 000 символов.

Формат выходных данных

Выходной файл должен содержать искомый палиндром. Если таких палиндромов несколько, выведите любой из них.

Примеры

stdin	stdout
anna	anna
perevorot	ror

Замечание

Идейно. Отрезки.

Задача 11F. Восстановление [3 sec, 512 mb]

Денис обнаружил ошибку в своей программе, которая должна удалять все символы из строки кроме “(” и “)”. Оказывается, некоторые скобки заменяются на что-то нечитаемое.

Теперь его заинтересовал вопрос, сколько различных правильных скобочных последовательностей могут являться результатом правильного алгоритма.

Формат входных данных

Единственная строка входного файла содержит строку из круглых скобок и знаков вопроса, где вопросами обозначены утраченные символы. Вопрос можно заменить на ровно одну любую скобку. Длина строки не превосходит 10 000, но может быть нечетной.

Формат выходных данных

Выведите одно число — количество различных скобочных последовательностей, удовлетворяющих шаблону Дениса, по модулю $10^9 + 7$.

Пример

stdin	stdout
(??()?	2

Подсказка по решению

До куда дошли и баланс.

Задача 11G. Восстановление [0.3 sec, 3 mb]

Денис обнаружил ошибку в своей программе, которая должна удалять все символы из строки кроме “(” и “)”. Оказывается, некоторые скобки заменяются на что-то нечитаемое.

Теперь его заинтересовал вопрос, сколько различных правильных скобочных последовательностей могут являться результатом правильного алгоритма.

Формат входных данных

Единственная строка входного файла содержит строку из круглых скобок и знаков вопроса, где вопросами обозначены утраченные символы. Вопрос можно заменить на ровно одну любую скобку. Длина строки не превосходит 10 000, но может быть нечетной.

Формат выходных данных

Выведите одно число — количество различных скобочных последовательностей, удовлетворяющих шаблону Дениса, по модулю $10^9 + 7$.

Пример

stdin	stdout
(??()?	2

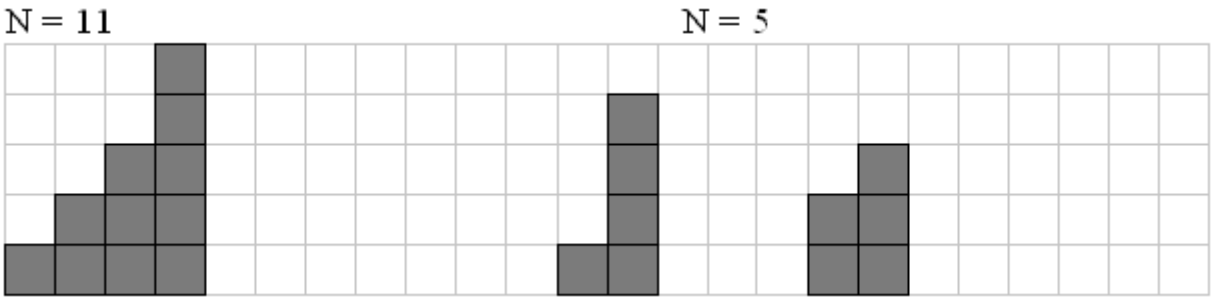
Замечание

Вы сдали предыдущую задачу?

Отлично, теперь храните только 1 или 2 строки динамики.

Задача 11H. Лестницы [0.1 sec, 3 mb]

У маленького мальчика есть набор из n кубиков ($5 \leqslant n \leqslant 500$). Из этих кубиков можно сложить различные лестницы. Лестницы имеют ступени различного размера, следующие в порядке возрастания этого размера (обратите особое внимание на то, что лестница не может иметь две одинаковые ступени). Каждая лестница должна иметь минимум две ступени, и каждая ступень должна состоять минимум из одного кубика. На рисунке приведены примеры лестниц для $n = 11$ и $n = 5$:



Найдите число q различных лестниц, которые маленький мальчик может построить ровно из n кубиков.

Формат входных данных

Число n.

Формат выходных данных

Число q.

Примеры

stdin	stdout
212	995645335

Замечание

Есть простое решение за $\mathcal{O}(n^2)$ времени и $\mathcal{O}(n)$ памяти.

Задача 11I. Шаблоны [1 sec, 3 mb]

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей, и т. д. Ваша задача — реализовать простейший алгоритм проверки шаблонов для имен файлов.

В этой задаче алфавит состоит из маленьких букв английского алфавита и точки (‘.’). Шаблоны могут содержать произвольные символы алфавита, а также два специальных символа: ‘?’ и ‘*’. Знак вопроса (‘?’) соответствует ровно одному произвольному символу. Звездочка ‘*’ соответствует подстроке произвольной длины (возможно, нулевой). Символы алфавита, встречающиеся в шаблоне, отображаются на ровно один такой же символ в проверяемой строчке. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить на символы строки таким образом, как описано выше. Например, строчки “ab”, “aab” и “beda.” подходят под шаблон “*a?”, а строчки “bebe”, “a” и “ba” — нет.

Формат входных данных

Первая строка входного файла определяет шаблон P. Вторая строка S состоит только из символов алфавита. Ее необходимо проверить на соответствие шаблону. Длины обеих строк не превосходят 10 000. Строки могут быть пустыми — будьте внимательны!

Формат выходных данных

Если данная строка подходит под шаблон, выведите YES. Иначе выведите NO.

Примеры

stdin	stdout
k?t*n kitten	YES
k?t?n kitten	NO

Дополнительные задачи

Задача 11J. Два шаблона [0.3 sec, 256 mb]

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей и т. д. Ваша задача — найти строку минимально возможной длины, которая подходит под два заданных шаблона.

Алфавит в этой задаче состоит из маленьких букв латинского алфавита и точки (‘.’). Шаблоны могут содержать любые символы алфавита, а также специальные символы ‘?’ и ‘*’. Под ‘?’ подходит любой символ алфавита, а под ‘*’ — любая строка символов алфавита (возможно, пустая). Под символы алфавита, встречающиеся в шаблоне, подходят только такие же символы алфавита. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отбразить в строку вышеуказанным способом. Например, строки «ab», «aab» и «beda.» подходят под шаблон «*a?», а строки «bebe», «a» и «ba» — нет.

Формат входных данных

Вам даны один или нескольких тестов. Первая строка ввода — количество тестов в нём. Каждый тест состоит из двух строк, содержащих шаблоны P_1 и P_2. Длина любого из шаблонов не превосходит 100 символов.

Формат выходных данных

Для каждого из тестов ответ задается одной строкой. Если строка, подходящая под оба шаблона, существует, выведите такую строку минимально возможной длины (если таких несколько, разрешается выводить любую). Иначе выведите «NO».

Пример

stdin	stdout
3 *k*tt*n* *i*e* haha hihi	iekttn NO

Замечание

Заметьте, бывают пустые строки.

Задача 11K. Интересное число [0.3 sec, 256 mb]

Для заданного числа n найдите наименьшее положительное целое число с суммой цифр n, которое делится на n.

Формат входных данных

Целое число n ($1 \leqslant n \leqslant 1000$).

Формат выходных данных

Выведите искомое число. Ведущие нули выводить не разрешается.

Пример

stdin	stdout
1	1
10	190

Замечание

$\Theta(n^3)$ заходить не должно. Но если вдруг, поделитесь историей успеха, пожалуйста.

Задача 11L. Таблицы Юнга [0.2 sec, 256 mb]

Нужно найти число способов расставить числа от 1 до N внутри диаграммы Юнга площади N так, чтобы числа внутри каждой строки и каждого столбца возрастали. Каждое число от 1 до N нужно использовать ровно один раз.

$1 \leqslant N \leqslant 50$

Формат входных данных

Число строк диаграммы Юнга k. Далее k длин строк: $a_1 \geqslant a_2 \geqslant \cdots \geqslant a_k$.
Число 1 должно оказаться в первой клетке первой строки.
Первый столбец диаграммы имеет высоту k. Высоты столбцов и длины строк, убывают.

Формат выходных данных

Одно целое число — количество способов расставить числа от 1 до N, где $N = \sum_{i=1}^{k} a_i$.

Пример

stdin	stdout
2 2 2	2
4 4 3 2 1	768

Задача 11M. Три типа скобок [0.1 sec, 256 mb]

Определим по индукции множество $\mathcal{T}$ *правильных скобочных последовательностей из трёх типов скобок*:

- $\varepsilon \in \mathcal{T}$ (пустая строка)
- $A \in \mathcal{T} \Rightarrow (A) \in \mathcal{T}$
- $A \in \mathcal{T} \Rightarrow [A] \in \mathcal{T}$
- $A \in \mathcal{T} \Rightarrow \{A\} \in \mathcal{T}$
- $A \in \mathcal{T}$, $B \in \mathcal{T} \Rightarrow AB \in \mathcal{T}$

Пусть теперь $\mathcal{T}_n$ — это множество правильных скобочных последовательностей из $2n$ символов — n открывающих и n закрывающих скобок.

Упорядочим элементы множества $\mathcal{T}_n$ лексикографически с некоторым порядком символов '`(`', '`)`', '`[`', '`]`', '`{`' и '`}`'.

По данным числам n и p, а также порядку, заданному на скобках, найдите p-ый в этом порядке элемент множества $\mathcal{T}_n$.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и p ($0 \leqslant n \leqslant 20$, $0 \leqslant p \leqslant 9 \cdot 10^{18}$). Скобочные последовательности нумеруются с нуля.

Во второй строке записаны шесть символов — '`(`', '`)`', '`[`', '`]`', '`{`' и '`}`' — в некотором порядке. Их порядок задаёт лексикографический порядок на множестве $\mathcal{T}_n$.

Формат выходных данных

В первой строке выходного файла выведите $2n$ символов без пробелов — p-ю правильную скобочную последовательность длины $2n$ из трёх типов скобок.

Если для данного n не существует p-я правильная скобочная последовательность, выведите в первой строке "`N/A`".

Примеры

stdin	stdout
1 0 ()}[{]	()
1 1 ()}[{]	[]
1 2 ()}[{]	{}
1 3 ()}[{]	N/A

Задача 11N. Японский кроссворд [0.1 sec, 256 mb]

Решение японских кроссвордов (они часто называются "Закраска числами") – очень популярное развлечение среди новых русских. В этом виде кроссвордов клетки должны закрашиваться чёрным или оставаться пустыми в соответствии с числами, заданными на сторонах прямоугольной сетки. Если всё сделано правильно, чёрные клетки складываются в картинку. Числа на сторонах сетки – длины последовательных отрезков закрашенных клеток в строке или столбце. Например, ключ "4 8 3" означает, что есть отрезки из четырёх, восьми и трёх заполненных клеток, в таком порядке, с хотя бы одной пустой клеткой между последовательными группами. Конечно, программисты не настолько умны, как новые русские, поэтому мы не просим вас решить весь кроссворд. Вашей задачей будет написать программу, которая находит как можно больше информации о каждой клетке одной строки кроссворда по некоторой уже имеющейся информации об этой строке.

Формат входных данных

Первая строка содержит длину строки L ($1 \leqslant L \leqslant 400$) и число групп последовательных клеток, которые должны быть закрашены K ($0 \leqslant K \leqslant L$). Вторая строка содержит K целых чисел – длины этих групп. Третья строка содержит L символов, описывающих текущую информацию о клетках этой строки (эта информация может быть получена анализом данных столбцов и других строк):

- `'.'` означает клетку, которая определённо пуста,
- `'X'` (латинская заглавная) означает клетку, которая определённо должна быть закрашена,
- `'?'` означает, что об этой клетке нет информации.

Формат выходных данных

Выведите строку, содержащую L символов, описывающих наиболее полную информацию о клетках строки в том же формате, что и во вводе, или слово "`Impossible`", если исходная информация противоречива, и никакая строка не может соответствовать входным данным.

Примеры

stdin	stdout
10 2 4 2 ?????.?X??	?XXX?.?X?.
9 0 ??????.X?	Impossible

Замечание

Анекдот в тему задачи: на IOI 2016 дали задачу timus:1508.